IT IS CLAIMED:

1. An encoding system adapted to encode data strings into codewords, the encoding system comprising:

a first memory portion adapted to store a dictionary of data strings and codewords corresponding to the data strings, wherein the dictionary is implemented as a balanced binary tree;

a second memory portion adapted to store a data string to be processed; and

an encoder adapted to receive from the second memory portion the data string to be processed, to determine if a codeword corresponding to a portion of the data string to be processed is stored in the dictionary and to output a codeword corresponding to a data string previously found in the dictionary if the codeword corresponding to the portion of the data string to be processed is not stored in the dictionary, wherein the encoder is further adapted to balance the dictionary.

2. The encoding system of claim 1, wherein the first and second memory portions comprise portions of a single memory.

3. The encoding system of claim 1, further comprising a buffer adapted to receive a variable length input and to outputs a fixed length output.

4. The encoding system of claim 1, wherein the encoder is adapted to balance the dictionary using an Adelson-Velskii and Landis (AVL) algorithm.

5. The encoding system of claim 1, wherein the dictionary is organized according to keys formed from a codeword corresponding to a set of characters of the data string to be processed and from an additional character of the data string to be processed.

6. The encoding system of claim 5, wherein the set of characters is received by the encoder before the additional character is received by the encoder.

7. The encoding system of claim 5, wherein the set of characters comprises a single character.

8. The encoding system of claim 5, wherein the set of characters comprises a plurality of characters.

9. The encoding system of claim 1, wherein the encoder is adapted to add a codeword to the dictionary if the codeword corresponding to the portion of the data string to be processed is not stored in the dictionary.

10. The encoding system of claim 1, wherein the encoder is adapted to delete codewords from the dictionary.

11. The encoding system of claim 1, wherein the encoder is adapted to balance the dictionary by rotating the dictionary.

12. The encoding system of claim 11, wherein rotating the dictionary comprises making right-right rotations.

13. The encoding system of claim 11, wherein rotating the dictionary comprises making left-left rotations.

14. The encoding system of claim 11, wherein rotating the dictionary comprises making right-left rotations.

15. The encoding system of claim 11, wherein rotating the dictionary comprises making left-right rotations.

16. The encoding system of claim 1, wherein the encoder comprises a plurality of state machines.

17. The encoding system of claim 1, wherein the encoder is adapted to determine if the dictionary is unbalanced.

18. The encoding system of claim 17, wherein the encoder is adapted to balance the dictionary if the dictionary is unbalanced.

19. A decoding system adapted to decode codewords into data strings, the decoding system comprising:

a memory adapted to store a dictionary of data strings and codewords corresponding to the data strings, wherein the dictionary is implemented as a balanced binary tree;

an input buffer adapted to receive and store a set of codewords to be processed; and

a decoder adapted to receive from the input buffer the set of codewords to be processed, to decode a first codeword into a first character string, to decode a second codeword into a second character string and to assign a third codeword to a combination of the first codeword and the second character string if a codeword corresponding to the combination of the first codeword and the second character string is not stored in the dictionary, wherein the decoder is further adapted to balance the dictionary.

20. The decoding system of claim 19, wherein the input buffer comprises a first-in, first-out buffer adapted to receive a fixed length input and to output a variable length output.

21. The decoding system of claim 19, wherein the decoder is adapted to balance the dictionary using an Adelson-Velskii and Landis (AVL) algorithm.

22. The decoding system of claim 19, wherein the dictionary is organized according to keys formed from the first codeword and the second character string.

23. The decoding system of claim 22, wherein the first codeword is received by the decoder before the second codeword is received by the decoder.

24. The decoding system of claim 22, wherein the first character string comprises a single character.

25. The decoding system of claim 22, wherein the first character string comprises a plurality of characters.

26. The decoding system of claim 19, wherein the decoder is adapted to delete codewords from the dictionary.

27. The decoding system of claim 19, wherein the decoder is adapted to balance the dictionary by rotating the dictionary.

28. The decoding system of claim 27, wherein rotating the dictionary comprises making right-right rotations.

29. The decoding system of claim 27, wherein rotating the dictionary comprises making left-left rotations.

30. The decoding system of claim 27, wherein rotating the dictionary comprises making right-left rotations.

31. The decoding system of claim 27, wherein rotating the dictionary comprises making left-right rotations.

32. The decoding system of claim 19, wherein the decoder comprises a plurality of state machines.

33. The decoding system of claim 19, wherein the decoder is adapted to determine if the dictionary is unbalanced.

34. The decoding system of claim 33, wherein the decoder is adapted to balance the dictionary if the dictionary is unbalanced.

35. An encoder adapted to operate with a first memory portion adapted to store a dictionary of data strings and codewords corresponding to the data strings, wherein the dictionary is implemented as a balanced binary tree, and a second memory portion adapted to receive and store a data string to be processed, the encoder comprising:

a first hardware state machine adapted to receive from the second memory portion the data string to be processed;

a second hardware state machine adapted to determine if a codeword corresponding to a portion of the data string to be processed is stored in the dictionary and to output a codeword corresponding to a data string previously found in the dictionary if the codeword corresponding to the portion of the data string to be processed is not stored in the dictionary; and

a third hardware state machine adapted to balance the dictionary.

36. The encoder of claim 35, wherein the first and second memory portions comprise portions of a single memory.

37. The encoder of claim 35, wherein the third hardware state machine is adapted to balance the dictionary using an Adelson-Velskii and Landis (AVL) algorithm implemented in hardware.

38. The encoder of claim 37, wherein the dictionary is organized according to keys formed from a codeword corresponding to a set of characters of the data string to be processed and from an additional character of the data string to be processed.

39. The encoder of claim 38, wherein the set of characters is received by the first hardware state machine before the additional character is received by the first hardware state machine.

40. The encoder of claim 38, wherein the set of characters comprises a single character.

41. The encoder of claim 38, wherein the set of characters comprises a plurality of characters.

42. The encoder of claim 35, wherein the second hardware state machine is adapted to add a codeword to the dictionary if the codeword corresponding to the portion of the data string to be processed is not stored in the dictionary.

43. The encoder of claim 35, wherein the second hardware state machine is adapted to delete codewords from the dictionary.

44. The encoder of claim 35, wherein the third hardware state machine is adapted to balance the dictionary by rotating the dictionary.

45. The encoder of claim 35, wherein the third hardware state machine is adapted to determine if the dictionary is unbalanced.

46. The encoder of claim 45, wherein the third hardware state machine is adapted to balance the dictionary if the dictionary is unbalanced.

47. A decoder adapted to operate with a memory adapted to store a dictionary of data strings and codewords corresponding to the data strings, wherein the dictionary is implemented as a balanced binary tree, and an input buffer adapted to receive and store a set of codewords to be processed, the decoder comprising:

a first hardware state machine adapted to receive from the input buffer the set of codewords to be processed;

a second hardware state machine adapted to decode a first codeword into a first character string, to decode a second codeword into a second character string and to assign a third codeword to a combination of the first codeword and the second character string if a codeword corresponding to the combination of the first codeword and the second character string is not stored in the dictionary; and

a third hardware state machine adapted to balance the dictionary.

48. The decoder of claim 47, wherein the third hardware state machine is adapted to balance the dictionary using an Adelson-Velskii and Landis (AVL) algorithm implemented in hardware.

49. The decoder of claim 47, wherein the dictionary is organized according to keys formed from the first codeword and the second character string.

50. The decoder of claim 49, wherein the first codeword is received by the first hardware state machine before the second codeword is received by the first hardware state machine.

51. The decoder of claim 49, wherein the first character string comprises a single character.

52. The decoder of claim 49, wherein the first character string comprises a plurality of characters.

53. The decoder of claim 47, wherein the second hardware state machine is adapted to delete codewords from the dictionary.

54. The decoder of claim 47, wherein the second hardware state machine is adapted to balance the dictionary by rotating the dictionary.

55. The decoder of claim 47, wherein the third hardware state machine is adapted to determine if the dictionary is unbalanced.

56. The decoder of claim 55, wherein the third hardware state machine is adapted to balance the dictionary if the dictionary is unbalanced.